Digital Certificate Support Updates From RACF and PKI Services for V2R1

South California RACF User Group

Wai Choi, CISSP
RACF/PKI Development
Poughkeepsie, NY

e-mail: wchoi@us.ibm.com

4/2013

Agenda

- **Digital certificates usage has been growing**
- **Continuous enhancements to fulfill customer requirements**
- **Two main components on certificate support:**
  - RACF: RACDCERT command and the R_datalib callable service
  - PKI Services
- **At the end of this presentation, you should have an understanding of the support from:**
  - RACF:
    - Enterprise PKCS#11
    - RACDCERT enhancement
  - PKI Services:
    - Enterprise PKCS#11
    - Extended Validation certificate
    - Granular certificate administration control
    - RFC and web interface currency
    - Other function enhancement

Enterprise PKCS#11 support

Overview

- **Problem Statement / Need Addressed**
  - Unlike the keys stored in the Public Key Data Set (PKDS), the keys stored in the Token Key Data Set (TKDS) are clear keys, not secure keys
  - “Secure Key” means that sensitive key material is always wrapped under a master key
  - The key generation capability from PKI Services is hampered by this clear key limitation
  - In Web Deliverable #12, ICSF supports secure key on TKDS
  - To enable the applications to use the secure key in TKDS, RACF, PKI Services and System SSL need to be updated accordingly

Overview

- **Solution**
  - RACDCERT can create secure key on a specified PKCS#11 token on TKDS during certificate creation
    - a label referencing this secure key will be stored in RACF in this format:

  '=' || <tkds-object-handle> || <spaces-padded-to-make-length-equal-64>

    - The '=' sign indicates it is a label to a TKDS secure key
  - the R_datalib callable service can return this label from RACF or from the PKCS#11 token
  - PKI Services can create secure key in TKDS during certificate creation and return a PKCS#12 package containing the secure key to the requestor
    - through new/updated System SSL APIs to envelop the secure key to be exported

  Note: RACDCERT EXPORT can not export any secure key neither from PKDS or TKDS

Overview

- **Benefit / Value**
  - This new support is intended to allow you to issue and use certificates with hardware-protected keys in the token
  - Provide better security on the key generation capability in PKI Services

Usage & Invocation

RACDCERT GENCERT / REKEY command and panels

- New sub keyword TOKEN is added to indicate the generation of secure TKDS key. For examples:
  - Generate a certificate with RSA key stored in a token called MY.PKCS11.TOKEN1 in TKDS

    RACDCERT GENCERT SUB(CN('Company A')) WITHLABEL('New RSA cert') RSA**(TOKEN(MY.PKCS11.TOKEN1))**

  - Generate a certificate with NISTECC key stored in a token called MY.PKCS11.TOKEN2 in TKDS

    RACDCERT GENCERT SUB(CN('Company A')) WITHLABEL('New ECC cert') NISTECC**(TOKEN(MY.PKCS11.TOKEN2))**

- Panels and corresponding help panels for GENCERT and REKEY will be updated to handle the new types.

R_datalib callable service

- For the existing private key type X'00000002' - ICSF key token label, if the first character is an '=' sign, it is a key token from the TKDS, otherwise it is from the PKDS.
- New private key types will be handled by functions DataGetFirst and DataGetNext
  - X'0000000B' RSA key token label in the TKDS
  - X'0000000D' ECC key token label in the TKDS
  - X'0000000E' DSA key token label in the TKDS

Usage & Invocation

- **PKI Services IKYSETUP (A REXX script to set up authorization for PKI)**
  - Update the script to pick the desire secure TKDS key to generate the CA certificate
    - key_type=8: Secure RSA in TKDS
    - key_type=9: Secure NISTECC in TKDS
    - key_type=10: Secure BPECC in TKDS

- **PKI Service configuration**
  - Specify T for the SecureKey entry in the PKI Services configuration file, pkiserv.conf to enable secure key generation

```
[SAF]
...
TokenName=PKISRVD.PKIToken
SecureKey=T
```

Note: SecureKey=F or absence of this keyword indicates clear keys or secure keys will be generated in the TKDS according to the installation configuration policy (see Migration & Coexistence Considerations)

Interactions & Dependencies

- **Software Dependencies**
  - ICSF web deliverable #12
- **Hardware Dependencies**
  - Crypto Express4S in EP11 mode on IBM zEnterprise server
- **Exploiters**
  - System SSL

Migration & Coexistence Considerations

- **With WD#12, ICSF checks a new RACF resource profile in the CRYPTOZ class to restrict the use of clear keys in the PKCS#11 services**
  - CLEARKEY.<token-label>

- **Insufficient access to CLEARKEY.SYSTOK-SESSION-ONLY will cause the failure on the generation of clear TKDS key from RACDCERT**

  Note: SYSTOK-SESSION-ONLY is an ICSF predefined temporary token name

- **Insufficient access to CLEARKEY.<token specified in pkiserv.conf> will cause unexpected result on key generation from PKI Services even no configuration update is made to have the new secure key support**
  - If Crypto Express4S is available, key generated is always a secure key
  - Otherwise, key generation fails

RACDCERT enhancement

Overview

- **Problem Statement / Need Addressed**
  - When importing a PKCS#12 or PKCS#7 certificate chain using the RACDCERT ADD command, only the end entity certificate can be named using the specified label, RACF generates labels for the rest of the certificates in the chain. It won't inform the user what labels have been added
  - There is no support in RACDCERT to display the certificate chain information
  - RACDCERT LIST and CHECKCERT only list certificate information on one single certificate

Overview

- **Problem Statement / Need Addressed**
  - Generate a certificate request, aka CSR, from RACDCERT GENREQ needs an existing certificate with a private key
    - The private key associated with the request is in fact the one associated with the certificate that GENREQ is based on
    - After the request is sent to the CA and CA issues the certificate, the certificate is imported back into RACF to replace the certificate that GENREQ has based on
  - Some clients have inadvertently deleted the original certificate once the request is generated, erroneously concluding that the certificate had no use
    - Since the certificate's private key is associated with the original certificate, deleting the certificate deletes the private key too
    - The certificate returned by the CA becomes completely useless, since it only contains the public key.

Overview

- **Solution**
  - Report to the user the labels used for all the certificates in the chain when the chain is added
  - Add function to display information on the certificates in the chain
  - Prevent the deletion of a certificate that has been used for generating a request; but also grant clients an override mechanism to delete it when needed

- **Benefit / Value**
  - Provide more comprehensive view on certificates in the system
  - Save customer time and money in obtaining a new certificate due to mistake

Usage & Invocation

RACDCERT ADD

- When RACDCERT ADD is successfully added a chain, a new message is issued for each added certificate on the label used
- For example: **RACDCERT ID(CHOI) ADD(<dataset contains the chain>) WITHLABEL('MyCert')**

  Messages issued:

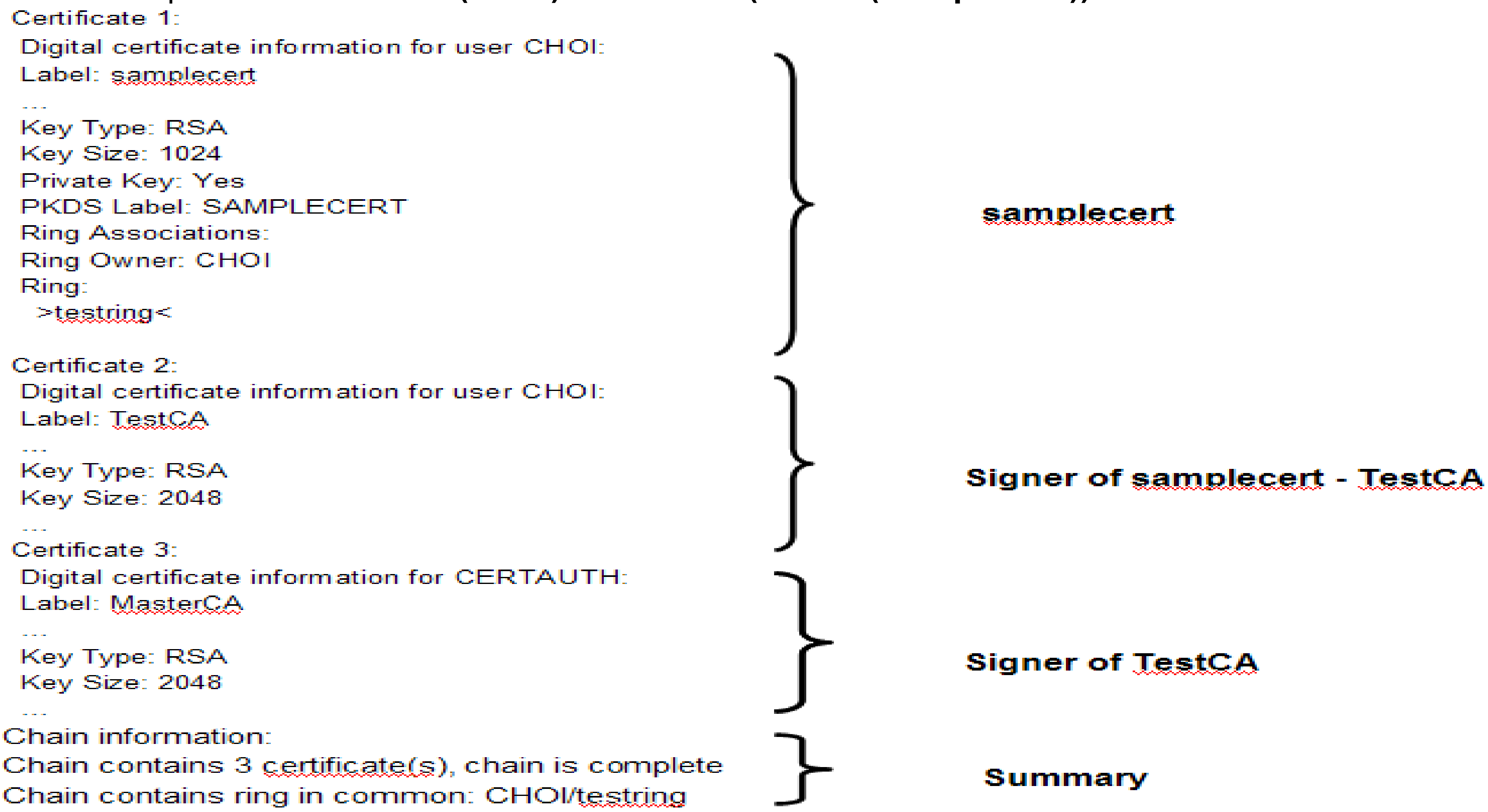
  Certificate with label 'MyCert' is added under ID CHOI
  Certificate with label 'LABEL00000002' is added under CERTAUTH
  Certificate with label 'LABEL00000003' is added under CERTAUTH

RACDCERT LISTCHAIN and CHECKCERT

- List certificate information on
  - a certificate owned by a user ID, SITE or CERTAUTH
  - its issuers' certificates owned by CERTAUTH or itself
  - a summary of the chain
- LISTCHAIN is to list certificates in RACF, while CHECKCERT is to list certificates in a dataset (which is going to be an input to the RACDCERT ADD)
- Similar output format

Usage & Invocation

– Example: **RACDCERT ID(CHOI) LISTCHAIN(LABEL('samplecert'))**



```
Certificate 1:
 Digital certificate information for user CHOI:
 Label: samplecert
 ...
 Key Type: RSA
 Key Size: 1024
 Private Key: Yes
 PKDS Label: SAMPLECERT
 Ring Associations:
 Ring Owner: CHOI
 Ring:
   >testring<

Certificate 2:
 Digital certificate information for user CHOI:
 Label: TestCA
 ...
 Key Type: RSA
 Key Size: 2048
 ...
Certificate 3:
 Digital certificate information for CERTAUTH:
 Label: MasterCA
 ...
 Key Type: RSA
 Key Size: 2048
 ...
Chain information:
Chain contains 3 certificate(s), chain is complete
Chain contains ring in common: CHOI/testring
```

samplecert

Signer of samplecert - TestCA

Signer of TestCA

Summary

PKI Extended Validation Certificate support

Overview

- **Problem Statement / Need Addressed**
  - An Extended Validation Certificate (EV) is an X.509 certificate issued according to a specific set of identity verification criteria
  - These criteria require extensive verification of the requesting entity's identity by the certificate authority (CA)
  - The certificates created by PKI Services do not support the relative distinguished names (RDN) that are required by Extended Validation certificates

Overview

- **Solution**
  - Support the RDNs needed for an EV certificate:
    - businessCategory (2.5.4.15) - Required
    - jurisdictionOfIncorporationCountryName (1.3.6.1.4.1.311.60.2.1.3) - Required
    - jurisdictionOfIncorporationStateOrProvinceName (1.3.6.1.4.1.311.60.2.1.2) - Optional
    - jurisdictionOfIncorporationLocalityName (1.3.6.1.4.1.311.60.2.1.1) - Optional

- **Benefit / Value**
  - Enables PKI Services administrators to potentially issue Extended Validation (EV) certificates which are of growing demand

Usage & Invocation

- **PKI Services configuration file (pkiserv.conf)**
  - Make sure the file has these new entries under the OID section:
    - BUSINESSCATEGORY=2.5.4.15
    - JURISDICTIONCOUNTRY=1.3.6.1.4.1.311.60.2.1.3
    - JURISDICTIONSTATEPROV=1.3.6.1.4.1.311.60.2.1.2
    - JURISTDICTIONLOCALITY=1.3.6.1.4.1.311.60.2.1.1

- **PKI Services template files (pkiserv.tmpl, pkitmpl.xml)**
  - Use the new template - 2-Year EV SSL Server Certificate

PKI Granular Administration Control support

Overview

- **Problem Statement / Need Addressed**
  - Need administration segregation inside the same PKI instance. For example, an administrator should be allowed to approve a server digital certificate, but not be allowed to approve a SCEP digital certificate

Overview

Solution

- Provide granular administration authorization control on requests and certificates based on the domain, action and the template
- A switch is provided to turn on this granular check
- A new class PKISERV is created for resources used by different types of administration functions
- If granular checking is on, these resources will be checked, in addition to the existing authority check on the administrative functions
- Example
  - READ access to

    MYDOMAIN.QUERYREQS.1YBSSL and
    MYDOMAIN.QUERYCERTS.1YBSSL
  - allow the administrator to perform QUERYREQS and QUERYCERTS on the requests and certificates respectively, created with the '1-Year PKI SSL Browser Certificate' template in domain named MYDOMAIN.

Overview

Benefit / Value

– Enable multiple PKI Services administrators to perform different actions on different types of certificates on different domains

Usage & Invocation

- **Set up new protection profiles in the new PKISERV class or update the IKYSETUP set up script**
  - PKI Services IKYSETUP (A REXX script to set up authorization for PKI)
    - Specify AdminGranularControl equals 1 to set up granular control
    - Provide the template nick names you want to act on and assign the corresponding administration groups for setting up new profiles in the PKISERV class

- **PKI Services configuration**
  - Specify T for the AdminGranularControl entry in the PKI Services configuration file, pkiserv.conf to enable granular control

  AdminGranularControl = T

PKI Currency

Overview

- **Problem Statement / Need Addressed**
  - Need additional information in Basic Constraints extension, which identifies
    - whether the certificate is a CA (required)
    - the maximum depth of the certification path (optional)
  - PKI Service only create the CA indication field, but not the path length value. Although it is optional, many customers would like to have that value set to control the number of CAs that can follow
  - PKI Services uses IBM HTTP Server Version 5.3 for z/OS which is based on the CERN web server. It will be replaced by the Apache version
  - The Websphere Application Server used by the JSP path is based on WAS 6.1 which is out of service

Overview

- **Solution**
  - Enable the PKI Services to optionally create the path length value in the Basic Constraints extension
  - Ensure PKI Services can run on the Apache HTTP server and WAS 7 and WAS 8

- **Benefit / Value**
  - Restrict a subordinate CA from signing another subordinate CA through the path length constraint value
  - Enable the customers to use new versions of web interfaces

Usage & Invocation

- **PKI Service configuration (pkiserv.conf)**
  - Specify T for the EnablePathLenConstraint entry to verify it meets path length constraint requirements and enables the setting of the pathLenConstraint field in the Basic Constraints extension of intermediate CA certificates created by this CA
  - Specify the length for the PathLength to be included in the Basic Constraints extension of intermediate CA certificates created by this CA

```
[CertPolicy]
...

EnablePathLenConstraint=T
PathLength=1
```

PKI Other enhancement

Overview

- **Problem Statement / Need Addressed**
  - There is no notification issued after the Certificate Revocation List (CRL) processing is done
  - When DB2 tables are used as the PKI Services backend stores, the customer should be able to add his own customized columns
  - This capability is restricted by the SQL processing done by the PKI Services daemon

Overview

- **Solution**
  - Enable PKI Services to optionally issue console message when CRL processing ends
    - IKYP044I CRL number crl-serial-number processing for CA domain ca-domain completed successfully
    - IKYP045I CRL number crl-serial-number processing for CA domain ca-domain failed
  - Issuance for the console messages is configurable to prevent unwanted console message from being issued
  - Enhance the SQL processing on the backend tables to facilitate DB2 customization

Overview

- **Benefit / Value**
  - A console message for CRL completion can act as a trigger for some automation processing, eg. CRLs can be saved for either legal reasons or a matter of policy
  - Enable customer who uses DB2 as backend stores to add his own customized columns in the tables which are used by PKI internal processing

Usage & Invocation

- **PKI Service configuration**
  - Specify 'file' for the **CRLWTONotification** entry in the PKI Services configuration file, pkiserv.conf to enable CRL notification
    - This keyword will be ignored if large CRL posting support is disabled (EnableLargeCRLPosting=F) or no http protocol CRL distribution point URI is defined

```
[CertPolicy]
...
CRLWTONotification=file
```

Session Summary

- **You should now have an understanding of the support from:**
  - RACF:
    - Enterprise PKCS#11
    - RACDCERT enhancement
  - PKI Services:
    - Enterprise PKCS#11
    - Extended Validation certificate
    - Granular certificate administration control
    - RFC and web interface currency
    - Other function enhancement

Appendix

- **Publication references**
  - Cryptographic Services PKI Services Guide and Reference (*SA22-7693*)
  - Security Server RACF Command Language Reference *(SA22-7687)*
  - Security Server RACF Administrator's Guide (SA22-7683)